INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS
Int. J. Numer. Anal. Meth. Geomech., **23**, 1141–1158 (1999)

SHORT COMMUNICATIONS

DYNAMIC SOIL–STRUCTURE INTERACTION ANALYSIS USING LANCZOS VECTORS WITH ADAPTIVE TIME INTEGRATION TECHNIQUE

R. VENUGOPALA RAO[1]*† AND N. S. V. KAMESWARA RAO[2]‡

[1] *National Institute of Rock Mechanics, Kolar Gold Fields-563 117, India*
[2] *Department of Civil Engineering, IIT Kanpur, 208 016, India*

SUMMARY

An analytical procedure to obtain the response of soil–structure interaction problems, *time domain* is described. The procedure makes use of *large* domain for descritization along with co-ordinate transformation using Lanczos vectors. The responses are obtained in time domain using an adaptive direct integration method. The scheme has the ability to estimate errors due to temporal discretization as well as co-ordinate transformation. The procedure has been applied to half-space problems and non-convex domains for validation of the scheme, and the scheme obeys causality condition in both the situations. The present method has all the advantages of time domain scheme which is local both in space and time with small computational effort. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: dynamics; soil–structure interaction lanczos vectors; adaptive technique; time domain analysis: finite element method transient analysis

1. INTRODUCTION

Analytical solutions for dynamic soil–structure interaction problems can be obtained for situations with highly idealized geometric representation of the problem, linear elastic material behaviour, and harmonic loading. Since soil exists as semi-infinite half-space, numerical methods have to incorporate the notion of infinity in the formulation. A simple approach to handle infinite domains is to consider a finite (computational) domain for discretization with an appropriate/approximating boundary conditions (energy absorbing) on the computational boundary, or a large domain with a boundary which reflects little energy.

Domain discretization (large size model) coupled with time integration (time domain approach) has been utilized to analyse dynamic soil–structure interaction problems, in the present study.

The Modelling consists of three stages—(1) spatial discretization, (2) Lanczos vector transformation and (3) temporal discretization.

* Correspondence to: R. Venugopala Rao, Numerical Modeling Division, National Institute of Rock Mechanics, Kolar Gold Fields-563 117, India. E-mail: nirm@giasbg01.vsnl.net.in
† Scientist
‡ Professor

CCC 0363–9061/99/111141–18$17.50

Received 27 February 1996
Revised 8 July 1998

The Finite Element Method is used for spatial discretization. The Lanczos vector transformation method, which is used to reduce the problem size, is described in Section 2. The temporal discretization needed to obtain the solution in time domain as implemented in this investigation is presented in Section 3. A detailed implementation of the method is described in Reference 1.

1.1. Application of FEM to dynamic problems

In this section the implementation of the Finite Element Method for spatial discretization is discussed. A displacement-type formulation uses nodal displacements u, v, w as primary variables. Variational formulation has been utilized to develop the procedure for spatial discretization. The discretization results in a system of second-order differential equations as

$$[M]\{\ddot{U}\} + [K]\{U\} = \{P\} \tag{1}$$

where M, C, K represent mass, damping and stiffness matrices of the system and P denotes the load vector. Here the vectors U and P are time dependent. In this present scheme, in the absence of boundary approximations, the damping terms will be absent. However to account for material damping, it is introduced in terms of Rayleigh damping as

$$[C] = \alpha[K] + p[M] \tag{2}$$

where α, and β are called the stiffness and mass proportional damping constants respectively. Incorporating the damping defined as above the equation of motion becomes

$$[M]\{\ddot{U}\} + [C]\{\dot{U}\} + [K]\{U\} = \{P\} \tag{3}$$

The above equation which describes the behaviour of the dynamical system is transformed to Lanczos space as explained in the next section. The transformed equations are used in time stepping scheme. The time histories of the generalized co-ordinates (of transformed domain) are projected on to the original space to obtain the displacement histories, etc. Here after the parenthesis around a variable, [], indicating that it represents a matrix quantity, will be omitted for concisseness.

2. LANCZOS METHOD

2.1. Introduction

Lanczos method is originally devised to extract some (not all) eigenvalues and eigenvectors of an eigensystem. It involves generation of orthonormal sequence of vectors known as Lanczos vectors iteratively from Krylov subspace. At each step the current vector is normalized with two previous vectors and build a tridiagonal matrix whose eigenvalues and eigenvectors are Rayleigh–Ritz approximations of the original system. This method (theoretically) terminates after n (order of the system) iterations. But, in practice, due to finite precision of numerical computation, the Lanczos vectors lose their orthogonality and may need re-orthogonalization frequently.

The Lanczos vectors which are orthonormal, can also be used to construct a co-ordinate transformation matrix to be used in the solution of dynamic equilibrium equations. Recently Wilson *et al.*[3] and Ibrahimbegovic *et al.*[4] developed a methodology to generate a set of orthogonal (Lanczos) vectors for use in co-ordinate transformation as an alternative to conventional mode super position method. Nour-Omid and Clough[5] showed that Lanczos vectors can

be used to solve structural dynamic problems in time domain. The basic methodology as implemented in this investigation in generating Lanczos vectors is described here briefly.

2.2. Lanczos algorithm

The Lanczos vectors are generated spanning the Krylov subspace defined by the sequence of vectors r, $(K^{-1}M)^1 r$, $(K^{-1}M)^2 r$,$(K^{-1}M)^3 r$, $\ldots$,$(K^{-1}M)^m r$, given the pair of matrices K and M and r being an arbitrary vector. The sequence of vectors converges to an eigenvector corresponding to the smallest eigenvalue of the generalized eigenvalue problem defined by the equation.

$$(K - \lambda M)z = 0 \tag{4}$$

In Lanczos algorithm, Gram–Schmidt orthogonalization[6] will be employed at each step to normalize the current vector with respect to matrix M and the two previous vectors of the Krylov sequence. The result will be a set of m orthonormal vectors.

Let $(q_1, q_2, \ldots, q_j)$ be j Lanczos vectors generated and $(j + 1)$th vector is to be found. Then q_{j+1} can be calculated by first computing a preliminary vector from the Krylov sequence as

$$\bar{r}_j = K^{-1} M q_j \tag{5}$$

Now the preliminary vector will be normalized with two preceding vectors q_{j-2}, q_{j-1} as

$$\bar{r}_j = r_j + \alpha_j q_j + \beta_j q_{j-1} + \gamma_j q_{j-2} + \cdots \tag{6}$$

where r_j is pure vector orthogonal to two previous vectors and α_j, β_j, and γ_j are the amplitudes of the previous vector contained in $\bar{r}_j$. The coefficients can be evaluated using the orthonormality of Lanczos vectors. Premultiplying the above equation by $q_j^{\mathrm{T}} M$, and noting the M orthonormality of the vectors, except first term all other terms on the right-hand side vanish, resulting in

$$\alpha_j = q_j^{\mathrm{T}} M \bar{r}_j \tag{7}$$

Similarly β_j can be evaluated as

$$\beta_j = q_{j-1}^{\mathrm{T}} M \bar{r}_j \tag{8}$$

Replacing $\bar{r}_j$ by its expression from equation (6) we get

$$\beta_j = q_{j-1}^{\mathrm{T}} M K - 1 M q_j = r_{j-1}^{\mathrm{T}} M q_j \tag{9}$$

Expanding $\bar{r}_{j-1}$ in terms of its pure vector r_{j-1} substituting in the transpose of equation (9) β_j becomes

$$\beta_j = q_j^{\mathrm{T}} M r_{j-1} + \alpha_{j-1} q_j^{\mathrm{T}} M q_{j-1} + \beta_{j-1} q_j^{\mathrm{T}} M q_{j-2} + \cdots \tag{10}$$

It is obvious that all terms except the first term vanishes on the right-hand side of the above equation vanish. Now q_j is the vector obtained by normalizing r_{j-1} with β_j, i.e.

$$q_j = \frac{1}{\beta_j} r_{j-1} \tag{11}$$

Substituting this expression in equation (10) results in

$$\beta_j = \frac{1}{\beta_j} r_{j-1} M r_{j-1} \tag{12}$$

Then the value of β_j will be

$$\beta_j^2 = r_{j-1} M r_{j-1} \tag{13}$$

In a similar procedure, it can be shown that γ_j and rest of the coefficient of the terms in equation (6) can be shown to zero.

In summary, the process of generating the Lanczos vectors can be expressed as follows:

Choose $q_0 = 0$, a starting vector r_0 and $\beta_1 = (r_0^{\mathrm{T}} M r_0)^{1/2}$. Then,

for $j = 1, 2, \ldots, m$ compute q_j as

$$q_j = \frac{r_{j-1}}{\beta_j} \tag{14}$$

$$\bar{r}_j = K^{-1} M q_j \tag{15}$$

$$r_j = \bar{r}_j - \alpha_j q_j - \beta_j q_{j-1} \tag{16}$$

$$\alpha_j = \bar{r}_j^{\mathrm{T}} M q_j \tag{17}$$

$$\beta_{j+1} = (r_j^{\mathrm{T}} M r_j)^{1/2} \tag{18}$$

2.3. *Starting vector*

The starting vector r_0, in general can be chosen arbitrarily. If these starting vector is related to loading amplitudes, we may expect that the modes which are contributing to the response will be included. Thus the starting vector is taken to be static displacement vector given by

$$r_0 = K^{-1} P \tag{19}$$

where P is the vector of loading amplitudes. Inclusion of the static displacement vector[3] avoids any possible need of applying static correction, because the static displacements are included in the co-ordinate transformation itself. Moreover, for small number of Lanczos steps, the basis vectors will be Rayleigh–Ritz approximations of scattered eigenvectors non-orthogonal to loading amplitudes in the Krylov sequence. The co-ordinate transformation matrix will contain approximations of eigenvectors of higher modes excited by the loading, resulting in a small number of basis vectors.

2.3.1. Selective re-orthogonalization. Lanczos vectors lose orthogonality due to the finite precision arithmetic and need re-orthogonalization. The aim of re-orthogonalization scheme is to prevent loss of orthogonality, i.e. to maintain a certain level of orthogonality among Lanczos vectors. We define level of orthogonality κ among the Lanczos vectors at jth step as

$$\kappa = \max_{1 \leqslant k \leqslant j-1} q_j^{\mathrm{T}} q_k \tag{20}$$

The full re-orthogonalization of q_j against all previous vectors maintains level of orthogonality κ, say around the level ε. However numerical results suggest[7,8] that semi-orthogonality i.e. $\kappa = \sqrt{(\varepsilon)}$ among the Lanczos vectors is sufficient to prevent spurious eigenvalues. This scheme

suggested by Simon[8] maintains semi-orthogonality among the Lanczos vectors. The monitoring of loss of orthogonality is done by updating an array consisting of levels of orthogonality. These levels of orthogonality are not computed explicitly, rather they are computed using a simple recurrence relation suggested by Simon.[8] Let ω's refer to the level of orthogonality between the Lanczos vectors referred by its subscripts, Then

$$\omega_{k,k} = 1 \quad \text{for } k = 1, \ldots, j$$
$$\omega_{k,k-1} = q_k^{\mathrm{T}} q_{k-1} \quad \text{for } k = 2, \ldots, j \tag{21}$$
$$\beta_{j+1}\omega_{j+1,k} = \beta_{k+1}\omega_{j,k+1} + (\alpha_k - \alpha_j)\omega_{j,k} + \beta_k\omega_{j,k-1} - \beta_j\omega_{j-1,k} + \psi_{j,k}$$

where $\psi_{j,k}$ is a random number to account for roundoff errors, It is suggested by Simon[8] that $\psi_{j,k}$ can be taken to be a random number with 0- mean and ε-standard deviation. Here it should be noted that the ω's are estimated or computed levels of orthogonality but found to be very close to the actual ones.[8]

At kth stage, update the recurrence relation for $\omega_{j+1,k}$ and find the vectors whose levels of orthogonality exceeded the semi-orthogonality level, i.e.

$$\text{for } k = 1, 2, \ldots, j \quad \text{find } j\text{'s such that } \|\omega_{j+1,k}\| \geqslant \sqrt{\varepsilon} \tag{22}$$

The vectors which satisfy the above equation are re-orthogonalized with current and next vector and corresponding ω's are reset to ε.

2.4. Number of vectors

The number of vectors required to obtain the results with desired accuracy can be obtained as follows. According Bayo and Wilson[9] and Coutinho *et al.*,[10] a measure of error introduced in the loading function in the transformed space can be used to gauge the accuracy of representation. It is expected that lower the error, higher is the accuracy. Let the transformation matrix be defined by the relation

$$U = \phi X \tag{23}$$

where X is the generalized displacement vector in the new co-ordinates space and $\phi = [\phi_1, \phi_2, \ldots, \phi_m]$, $m < n$ is the co-ordinate transformation matrix. The dynamic response $U(t)$ is now approximated by a linear combination of generalized displacements $X(t)$. Substituting for U in equation (3), we get

$$\bar{M}\ddot{X} + \bar{C}\dot{X} + \bar{K}X = \bar{P}(t) \tag{24}$$

where $\bar{M}$,$\bar{C}$, $\bar{K}$ are generalized mass, damping, and stiffness matrices and $\bar{P}(t)$ in the generalized load vector, respectively, and are given by the following equations:

$$\bar{M} = \phi^{\mathrm{T}} M \phi \tag{25}$$
$$\bar{C} = \phi^{\mathrm{T}} C \phi \tag{26}$$
$$\bar{K} = \phi^{\mathrm{T}} K \phi \tag{27}$$
$$\bar{P}(t) = \phi^{\mathrm{T}} P(t) \tag{28}$$

Premultiplying equation (25) by $M\phi$ and considering the spatial variation only, the load representation P_m in original space incorporated in the analysis can be expressed as

$$P_m = \sum_{j=1}^{m} M\phi\bar{p}_j \tag{29}$$

where p_j is jth participation factor and is given by

$$\bar{p}_j = \phi_j^{\mathrm{T}} P \tag{30}$$

Now the error E in loading representation can be expressed as

$$E = P - P_m \tag{31}$$

It is convenient to express the error in a suitable norm relative to the original load as

$$e = \frac{\|P^{\mathrm{T}}E\|}{\|P^{\mathrm{T}}P\|} \tag{32}$$

The relative error norm e will lie between 1 (for $m = 0$ i.e. 0 vectors and used) and 0 (for $m = n$ i.e. all vectors are used). A suitable value of error tolerance (relative, viz. in terms of percentage) may be specified and the Lanczos vectors are generated till the relative error norm e falls below the error tolerance of the load vectors.

2.5. Co-ordinate transformation matrix

Co-ordinate transformation matrix can be derived as follows:

1. Triangularize the stiffness matrix K, i.e.

$$K = L^{\mathrm{T}}DL \tag{33}$$

   where L and D are the lower triangular and diagonal matrices, respectively.
2. Choose static displacement vector as r_0, r as explained in the previous section.
3. Generate m Lanczos vectors which are orthonormal (apply partial re-orthogonalization where necessary). Check whether e relative error norm of load representation is less than error tolerance.
4. Project the stiffness matrix into Lanczos subspace as

$$K_m = Q^{\mathrm{T}}KQ, \quad Q = [q_1, q_2, \ldots, q_m] \tag{34}$$

5. Solve the eigenproblem in Lanczos subspace

$$K_m Y = I_m Y \Omega_m, \quad Y = [y_1, y_2, \ldots, y_m] \tag{35}$$

6. Compute co-ordinate transformation matrix

$$\phi = QY \tag{36}$$

3. DIRECT INTEGRATION

3.1. Introduction

Dynamic analysis of engineering systems requires temporal and often spatial discretization (for continuous systems). The spatial discretization results in a system of coupled differential

equations such as equation (3), describing the physical behaviour. These equations, in principle, can be solved by direct integration. The direct integration method will be efficient in case the excitation is either transient or periodic consisting of combination of several frequencies. For non-linear dynamic problems the direct integration methods are the only tools readily applicable. A direct integration method with desirable properties of damping out higher frequencies without loosing order of accuracy has been described here.

3.2. Θ *method*

The Θ method of Hoff and Pahl[11] which has been used in this investigation, is described briefly.

The dynamic equilibrium may be described by the original (equation (3)) or by transformed equation (24) with the given initial conditions.

If x_n, v_n, a_n and P_n are the displacement, velocity, acceleration and force vectors at nth step, respectively; then the corresponding quantities at $(n+1)$th step (with step size h) can be obtained as

$$a_{n+1} = a_n + \delta a \tag{37}$$

$$v_{n+1} = v_n + ha_n + (1{\cdot}5 - \theta)h\delta a \tag{38}$$

$$x_{n+1} = x_n + hv_n + 0{\cdot}5h^2 a_n + \frac{1}{4\theta_1^2} h^2 \delta a \tag{39}$$

$$M_m \delta a = \bar{P}_{n+1} - Ma_n - C\bar{v}_{n+1} - K\bar{x}_{n+1} \tag{40}$$

where

$$M_m = M + (1{\cdot}5 - \theta_1)hC + \frac{1}{4\theta_1^2} h^2 K$$

$$\bar{P}_{n+1} = P_n + \theta_0(P_{n+1} - P_n)$$

$$\bar{v}_{n+1} = v_n + \theta_1 ha_n$$

$$\bar{x}_{n+1} = x_n + \theta_1 hv_n + 0{\cdot}5h^2 a_n$$

$$\text{with } 0{\cdot}95 \leqslant \theta_1 \leqslant 1{\cdot}0$$

The matrices K, C, M are the stiffness, damping and mass matrices of the system. P_n is the load vector at nth step. These quantities refer to the *transformed* system.

Since this is a second-order method, the step size adjustment can be incorporated using Richardson extrapolation.[12] That is, compute the solution at time $t + h$ with step size h, and $h/2$. Then the difference of the resulting solutions gives an estimate of error as

$$\text{err} = \| \tfrac{7}{8} (x_h - x_{h/2})^{1/3} \| \tag{41}$$

where err is a measure of local truncation error, x_h and $x_{h/2}$ are displacement vectors at time $t + h$ with step sizes h and $h/2$, respectively. The symbol $\|\cdot\|$ here denotes a maximum norm.

3.3. Θ *Method with post-processing*

This method differs from the earlier one only in computation of local error. The post-processing of solution is done at the end of each step to estimate the error and consequently the step size required for prescribed error tolerance. In this method acceleration is assumed to be constant in each step. If a continuous (viz. linear) valued function of time is utilized for approximation of acceleration, more accurate results could be obtained (see Reference 13). If a^* is the acceleration at any time t during the nth step, it can be expressed as

$$a^* = a_n + \frac{t}{h}\,\delta a \tag{42}$$

where $t_n \leqslant t \leqslant t_{n+1}$ and $h = t_{n+1} - t_n$.

Now the error in acceleration during the nth step can be written as

$$e = a^* - a \tag{43}$$

where a is the average acceleration vector during the nth step and can be expressed as

$$a = \left(1 - \frac{1}{2\theta_1^2}\right) a_n + \frac{1}{2\theta_1^2}\, a_{n+1} \tag{44}$$

The local truncation error in displacements is obtained by integrating the error in accelerations two times. A norm of error in displacements ||err|| can be expressed as

$$\|\mathrm{err}\| = \left(\frac{1}{6} + \frac{1}{4\theta_1^2}\right) h^2 \|\delta \mathrm{a}\| \tag{45}$$

Since the above norm is an absolute quantity and problem dependent specifying an absolute value is not feasible always. A relative error norm $\|\mathrm{err}_{\mathrm{rel}}\|$ which is the ratio of error norm ||err|| and norm of the displacements at the current step is used for step size correction, i.e.,

$$\|\mathrm{err}_{\mathrm{rel}}\| = \frac{\|\mathrm{err}\|}{\|x\|} \tag{46}$$

Select the step size economically such that for each step, the local error roughly equals the prescribed error tolerance. This can be achieved as follows:

Define two parameters γ_1 and γ_2 such that $0 \leqslant \gamma_1 \leqslant 1$ and $\gamma_2 \geqslant 1$ and

$$\gamma_1 \varepsilon \leqslant \|\mathrm{err}_{\mathrm{rel}}\| \leqslant \gamma_2 \varepsilon \tag{47}$$

Accept the step if equation (47) is satisfied. In practice the step size adjustment is carried out if

1. Error is more than the upper limit of the tolerance ($\gamma_2 \varepsilon$) or
2. Error falls below the lower limit of tolerance ($\gamma_1 \varepsilon$) for *fixed number* of times (say 5–10) consecutively. This is to avoid costly factorization frequently needed whenever step change is made.

The new step size can be calculated as

$$h_{\mathrm{new}} = h * \left(\frac{\varepsilon}{\mathrm{err}_{\mathrm{rel}}}\right)^{1/3} \tag{48}$$

4. VALIDATION OF THE MODEL

4.1. Introduction

The set of problems considered by Von Estorff *et al.*[14] are chosen for validation purposes. The problem set consist of (1) A half-space and (2) A half-space with a vertical trench. The problems were modelled in two dimensions and are subjected to a line load. To simulate transient loading a fast decaying Ricker's wavelet[14] has been chosen. The effect of model parameters such as number of Lanczos vectors (Section 2.4) considered, size of the mesh, element size (number of elements) have been studied and reported here. The implementation of the model for these problems has been discussed in the following sections. The effect of different (from that of the problem to be solved) spatial distributions of the load is also considered for the generation of Lanczos vectors. It will be useful to understand the effect of spatial variation of the load on the accuracy, and to evolve a simpler computation model for infinite domain independent of the loading. A value of $\theta_1 = 0{\cdot}988$ which produces small numerical damping is used in time integration. The value of damping parameters α and β are taken to be 4 and 2 per cent, respectively.

4.2. Example 1. Elastic half-space

In this example an elastic half-space (see Reference 14) subjected to a Ricker's wavelet (transient excitation) in vertical and horizontal directions is investigated. The displacement histories at points 'A' and 'B' which are located at a distance of 20 and 50 m on the surface from the load are monitored.

A schematic diagram of the problems considered are shown in Figures 1 and 2. Only one-half of the domain is discretized for vertical loading and full domain is used for horizontal loading.

The parameters of the soil are as follows—Young's Modulus $= 2{\cdot}66 \times 10^5$ kN/m^2, Poisson's ratio $= 0{\cdot}33$, and density $= 2{\cdot}0$ t/m^3. The load function $f(t) = (1 - 2r^2)\mathrm{e}^{-r^2}$, where $r = \pi t - 3$.

4.3. Size of the model

To investigate the effect of the size of the model (computational domain), four models of sizes 400×400 m^2 (small), 400×600 m^2 (small), 600×600 m^2 (medium) and 800×800 m^2 (large) are

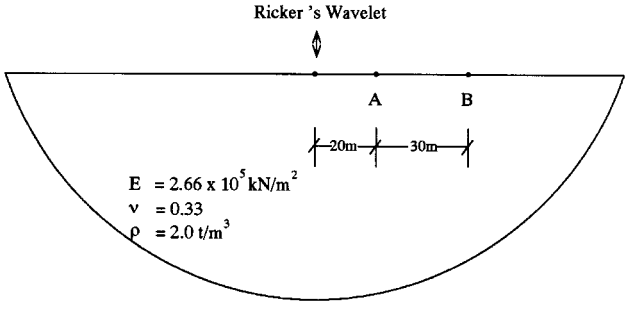

Figure 1. Half-space loaded by vertical Ricker's wavelet

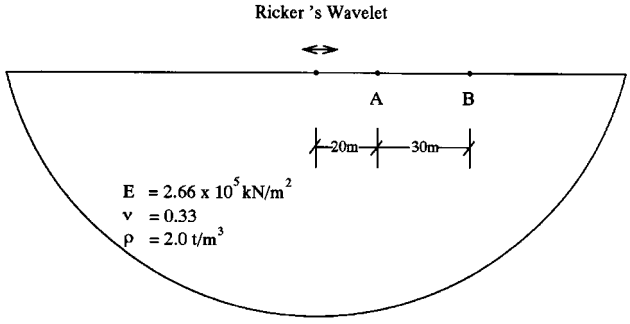

Figure 2. Half-space loaded by horizontal Ricker's wavelet

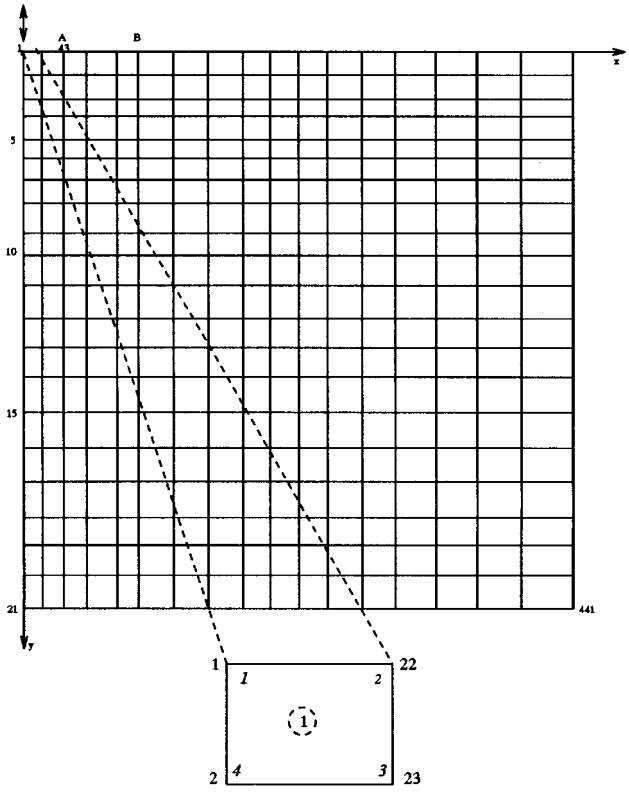

Figure 3. A typical FE mesh

considered in case of vertical loading (Figure 3). Number of elements in each direction are taken to be 20. The time step is monitored carefully so that *Courant* criterion[15] is not violated. Lumped mass method is adopted for the analysis. Time integration is carried out for 3 s. The vertical displacements at point 'A' obtained with these models are shown in Figure 4. The results obtained

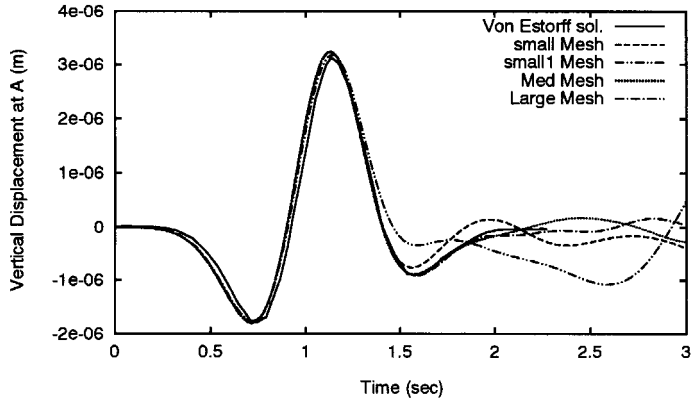

Figure 4. Effect of mesh size on vertical displacements

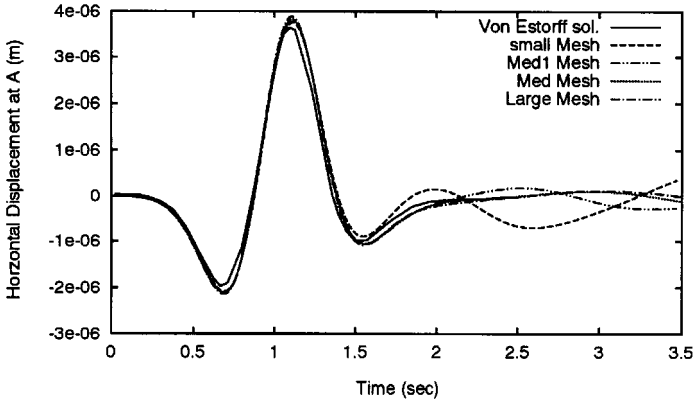

Figure 5. Effect of mesh size on horizontal displacements

with the models compare very well with those of Von Estorff *et al.*[14] obtained using boundary element methods. Only in small models, the displacements beyond 1·5 s do not match well, which could be due to the influence of reflected waves. The method also been applied wherein external loading is periodic. The time history plots were observed for sufficiently long duration of time (approximately 10 periods of loading) for increase in the amplitudes and found the amplitudes stable. It conclusively proves that those boundary reflections, if present any are insignificant.[1]

Similar study has been carried out with models $800 \times 400\ m^2$ (small), $1200 \times 400\ m^2$ (small 1), $1200 \times 600\ m^2$ (medium) and $1600 \times 800\ m^2$ (large) for horizontal excitation. The number of elements used in horizontal and vertical directions are 40 and 20, respectively. Horizontal displacements at point 'A' obtained with these three models are shown in Figure 5. The results agree very well with those reported by Von Estorff *et al.*[14] except for those obtained with small meshes at times beyond 1·5 s.

4.4. No. of Elements

In this section the effect of number of elements (element size) on the accuracy is reported. The computational domain of $800 \times 800\ m^2$ size is discretized with varying no. of elements from 20 to 40 in both the directions. The vertical displacement history at point 'A' subjected to a vertical load is obtained with these three mesh schemes and plotted in Figure 6. It can be observed that all the three models produced results of almost same accuracy and agree well with those reported by Von Estorff *et al.*[14]

4.5. Number of Lanczos vectors

To study the effect of *number of Lanczos vectors* considered for co-ordinate transformation, the error norm in load representation as defined by equation (32) is computed. Both absolute and relative norms are shown in Figure 7. The effect of number of Lanczos vectors on the accuracy of the results are shown in terms of vertical displacement history at point 'A' in Figure 8. The effect can be seen in the case of horizontal displacement of half-space loaded by horizontal Ricker's wavelet in Figure 9. As expected error norm in load representation reduces rapidly with increase in number of Lanczos vectors. Similarly accuracy of the results improves rapidly as number of vectors increase (reported in the following sections). It can be observed that 20 Lanczos vectors could produce solutions of desired accuracy.

4.5.1. Computational time. To assess the computational resources required for the first two stages of the modelling scheme, CPU times are logged in and analysed in this section. The first two stages of the scheme namely the spatial discretization and Lanczos vector transformation can be divided into the following steps: (1) FE Assembly; (2) Lanczos vector generation; (3) Transformation matrix generation; (4) Writing transformed system matrices and Lanczos vectors (I/O) to disk. CPU times required for each of these steps obtained on DEC ALPHA 3000s600 are given in Table I. The same information is shown in Figure 12. CPU time required for step 1 (FE assembly) remains constant with number of Lanczos vector because this step is independent of Lanczos vector generation. The CPU time required for each step per vector (slope of the lines in Figure 12) remains same for steps 2 and 4, whereas for step 3 which involves Ritz vector computation (equation (35)), the CPU time increases. However, it can be observed that total CPU time per vector (column (7)) does not vary much.

Table I. CPU times for various stages of computation (in seconds)

No. of vectors (1)	FE assembly (2)	Lanc. vect generation (3)	Transform matrix (4)	I/O (5)	Total (6)	Total time per vector (7)
5	0·5232	0·3867	0·1376	0·3670	1·4145	0·2829
10	0·5231	0·6900	0·3338	0·7837	2·3306	0·2331
20	0·5241	1·2171	0·8042	1·5460	4·0914	0·2046
30	0·5339	1·8954	1·8641	2·3327	6·6261	0·2209
50	0·5348	3·6246	4·5193	3·9138	12·5925	0·2519
80	0·5290	7·5308	10·2294	6·3624	24·6516	0·3081

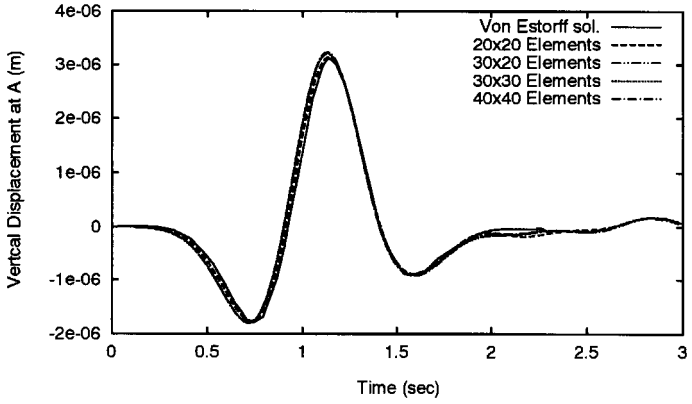

Figure 6. Effect of number of elements

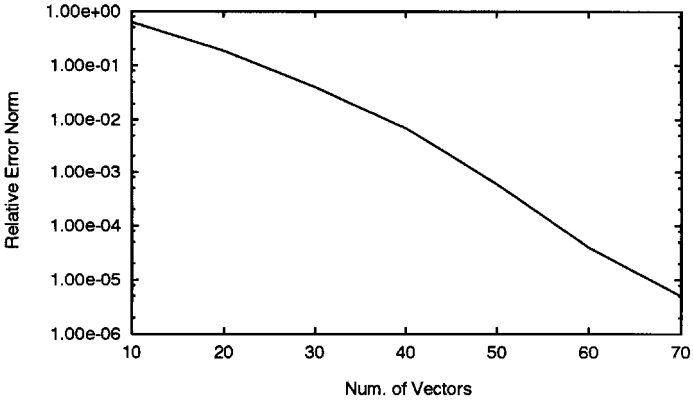

Figure 7. Effect of number of vectors on relative error norm

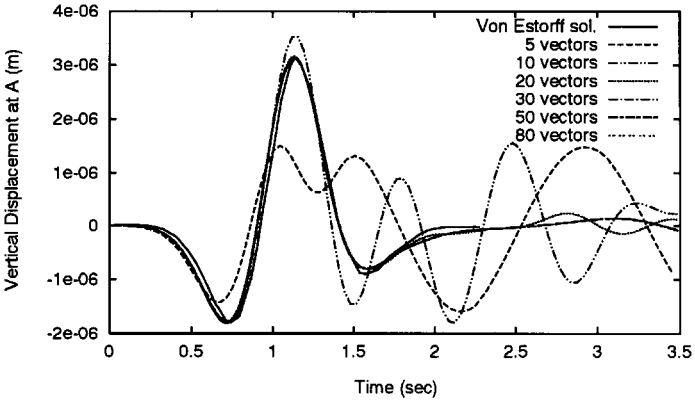

Figure 8. Effect of number of vectors on vertical displacement

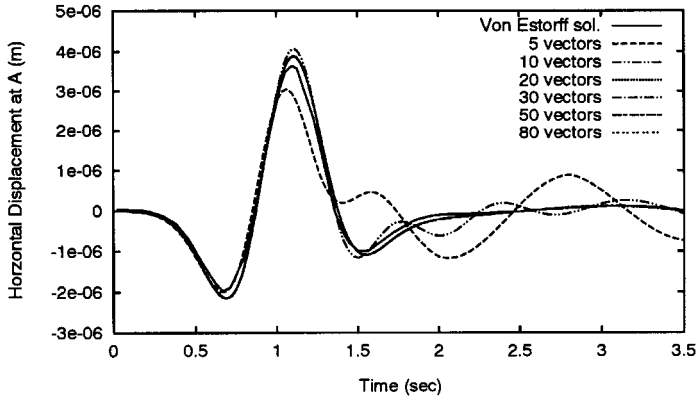

Figure 9. Effect of number of vectors on horizontal displacement

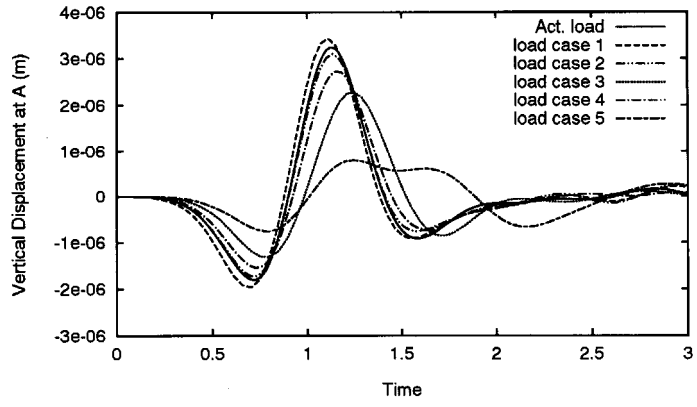

Figure 10. Effect of spatial variation of the load with 20 vectors

4.6. Spatial variation of the load

In this section different loading schemes are used to generate the Lanczos vectors only namely;

case 1: load at 10 m right (node 22) of original position (node 1),
case 2: load at 10 m down (node 2) the original position (node 1),
case 3: load at 50 m right (node 85) of original position (node 1),
case 4: load at 50 m down (node 4) the original position (node 1),
case 5: load at 100 m right (node 127) of original position (node 1).

Large mesh with 20 elements in each direction is utilized for the study. Displacement histories at points 'A' and 'B' are obtained with 20 and 40 Lanczos vectors. Vertical displacement history at point 'A' for vertical loading with 20 and 40 number of Lanczos vectors are shown in Figures 10 and 11. It can be observed that results obtained with loading cases 2 and 4 match well; where as the ones obtained with other schemes do not match well with those obtained with actual loading

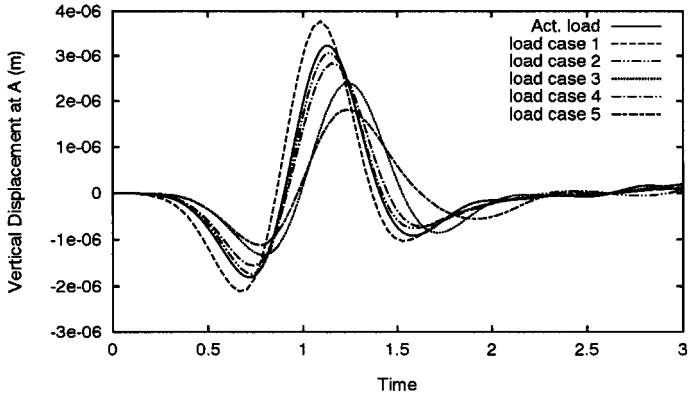

Figure 11. Effect of spatial variation of the load with 40 vectors

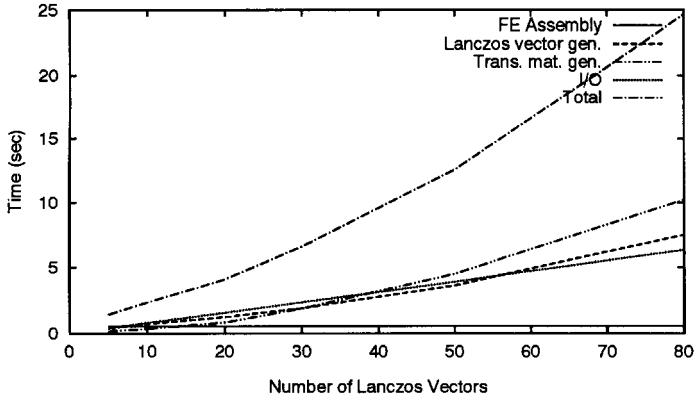

Figure 12. CPU time for various stages of computation

scheme; It may be observed that if the loading scheme used for generation of Lanczos vectors is spatially farther (loading cases 3 and 5) the method yields poor results, however an increase in the number of vectors improves the accuracy. If the loading considered is close to the observation region, results obtained will be overestimated (loading case 1).

It may be inferred that spatially different loading schemes can be used to generate Lanczos vectors and system matrices keeping in mind the following points.

1 The loading scheme should not be very different (like loading cases 3 and 5) from actual loading.
2 The loading scheme should not close to the observation region (like loading case 2).

4.7. Example 2. Elastic half-space with a vertical trench

This example helps in understanding the behaviour and applicability of the scheme for non-convex domains. A schematic diagram of half-space with vertical trench adjacent to the load is shown in Figure 13. The geometry of the problem is same as in earlier example except that

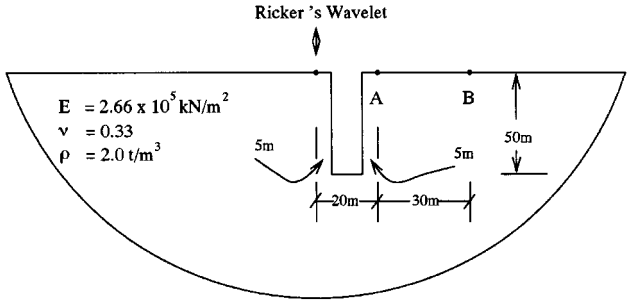

Figure 13. Half-space with trench loaded by vertical Ricker's wavelet

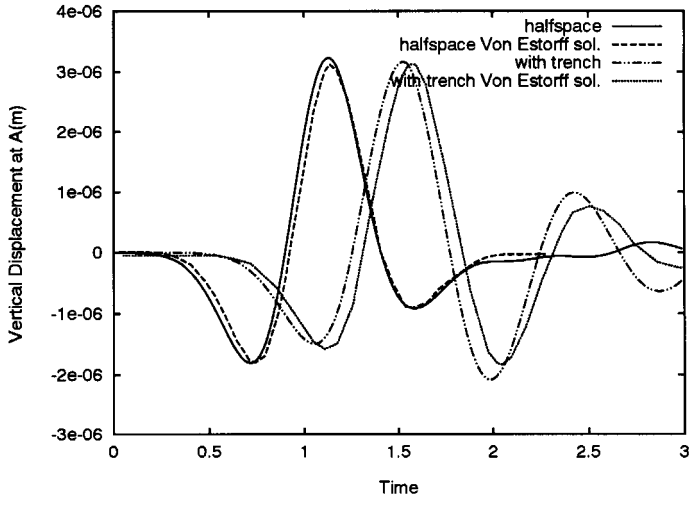

Figure 14. Effect of trench on vert. displacements due to vert load

a vertical trench of 50 m deep and 10 m wide is located at 5 m to right of the load. In this example also both vertical and horizontal loading schemes are considered. The displacement history at point 'A' for both vertical and horizontal loading are shown along with those reported by Von Estorff *et al.*[14] in Figures 14 and 15. It can be observed that the results obtained by the present model agree well with those obtained by Von Estorff *et al.*[14]

It is important to mention that the solutions obey the causality conditions in all the situations investigated. This could be verified directly from the displacement histories reported. Moreover solutions obtained by Von Estorff *et al.*[14] are found to satisfy this requirement and the present solutions agree well with them.

5. CONCLUSION

Based on the numerical investigation carried out the following conclusions can be drawn.

1. A large mesh coupled with the Lanczos vector transformation technique can represent semi infinite soil mass for computational purposes.

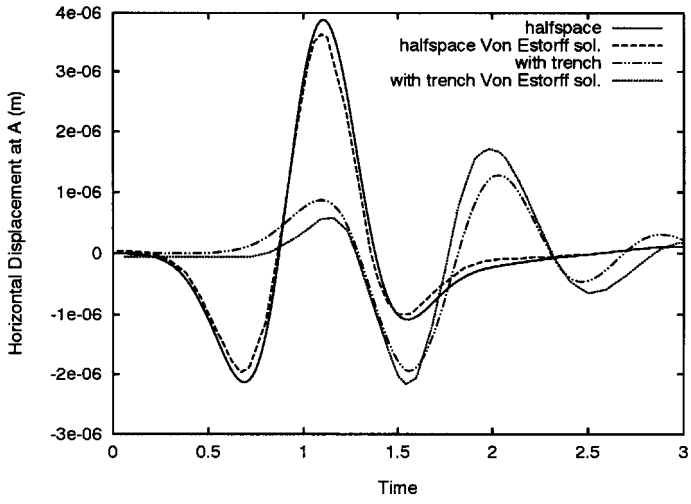

Figure 15. Effect of trench on horizontal displacements due to horizontal load

2. The present numerical scheme can effectively handle transient loading in time domain.
3. The model can estimate and control errors due to both co-ordinate transformation and temporal discretization.
4. The error introduced due to co-ordinate transformation decreases exponentially (see Figure 7) with an increase in number of Lanczos vectors. The computational cost per vector remains almost constant and a desired level of accuracy can be obtained with a sufficient number of Lanczos vectors.
5. The system (transformed) matrices generated with a particular spatial variation of the load can be used to analyse other loading schemes if the loading schemes are similar (see Section 4.6).
6. A transformed model with few degrees of freedom (about 20–40) can simulate adequately a physical model with a large (about 840) degrees of freedom.
7. The present model can effectively analyse non-convex domains (such as vertical trench problems).
8. The solutions obtained with the present model obey the causality condition.
9. With relatively small computational effort semi-infinite half-space problems with transient loading can be analysed in *time domain*.
10. It is important to emphasize that the present scheme is local both in space and time.

REFERENCES

1. R. Venugopala Rao, '*Time domain analysis of three-dimensional soil–structure interaction problems*'. Ph.D. Thesis, Indian Institute of Technology, Kanpur, (1995).
2. C. Paige, 'Error analysis of the Lanczos algorithm for triadiagonalizing a symmetric matrix', *J. Inst. Math. Appl.*, **18,** 341–349 (1976).
3. E. L. Wilson, M. Yuan and J. M. Dickens, 'Dynamic analysis by direct superposition of ritz vectors', *Earthquake Engng. Struct. Dyn.* **7,** 405–411 (1979).

4. A. Ibrahimbegovic and C. Harn, 'Ritz method for dynamic analysis of large discrete linear systems with non-proportional damping', *Earthquake Engng. Struct. Dyn.* **19,** 877–889 (1990).
5. B. Nour-Omid and R. W. Clough, 'Dynamic analysis of structures using Lanczos co-ordinates', *Earthquake Engng. Struct Dyn.* **11,** 565–577 (1984).
6. J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, Vol. 1 and 2, Birkhauser, Boston Inc., U.S.A. (1985).
7. B. N. Parlett and D. S. Scott, 'The Lanczos algorithm with selective orthogonalization', *Math. Comput.* **33,** 217–238 (1979).
8. H. D. Simon, 'The Lanczos algorithm with partial reorthogonalization', *Math. Comput.* **42,** 115–142 (1984).
9. E. P. Bayo and E. L. Wilson, 'Use of ritz vectors in wave propagation and foundation response', *Earthquake Engng. Struct. Dyn.* **11,** 499–505 (1984).
10. A. L. G. A. Coutinho and L. Landau, 'The application of the Lanczos mode superposition method in dynamic analysis of offshore structures', *Comput. Struct.* **25,** 615–625 (1987).
11. C. Hoff and P. J. Pahl, 'Development of an implicit method with numerical dissipation from a generalized single step algorithm for structural dynamics', *Comput. Meth. Appl. Mech. Engng.*, **67,** 367–385 (1988).
12. J. Stoer and R. Bulrsch (1980). *Introduction to Numerical Analysis*, Springer, New York.
13. L. F. Zeng and N. E. Wiberg, 'A posteriori local error estimation and adaptive time-stepping for newmark integration in dynamic analysis', *Earthquake Engng. Struct. Dyn.* **21,** 555–571 (1992).
14. O. Von Estorff, A. L. Pais and E. Kausel, 'Some observations on time domain and frequency domain boundary elements', *Int. J. Numer. Meth. Engng.*, **29,** 785–800 (1990).
15. R. L. Higdon, 'Absorbing boundary conditions for acoustic and elastic waves in stratified media', *J. Comput. Phys.* **101,** 386–415 (1992).

